УДК 519.876.5

АМЕР ТАХСИН САЛАМЕХ АБУ-ДЖАССАР, аспирант,
Е. В. ДУРАВКИН, канд. техн. наук,
Г. Н. ЖОЛТКЕВИЧ, д-р техн. наук, ХНУ имени В. Н. Каразина

ОБ ОДНОЙ ЛОГИЧЕСКОЙ МОДЕЛИ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ИНФОРМАЦИИ НА БАЗЕ СЕТЕЙ ПЕТРИ С МОДИФИЦИРОВАННОЙ ДИНАМИКОЙ

У статті розглянуто підхід до моделювання механізмів доступу до інформаційних ресурсів у розподіленій системі обробки інформації. Запропонованих підхід базується на модифікації апарату мереж Петрі та засобів його аналізу (дерева досяжності). Використання розроблених механізмів аналізу розподілених систем дозволить виявляти конфліктні ситуації при використанні ін формаційних ресурсів, а також вказувати на послідовність станів системи, що приводять до таких ситуацій.

1. Постановка проблемы

Увеличение сложности задач, решаемых с помощью информационно-вычислительных систем, приводит к необходимости объединения различных вычислительных комплексов в единую систему. Сложность построения систем данного класса связанна с комплексом проблем. Среди них необходимо отметить распределенность обработки информации с сохранением ее целостности и использование ограниченного набора ресурсов [1]. Одной из наиболее важных задач является обеспечение эффективного распределения ресурсов между программными процессами системы. Качество решения данной задачи во многом определяет эффективность функционирования системы в целом. Следовательно, разработка качественного программного обеспечения для систем данного класса не возможна без предварительной разработки некой формальной модели.

2. Анализ публикаций, постановка задачи

Сети Петри являются наиболее подходящим формализмом, предназначенным моделирования и анализа различных параллельных и распределенных систем обработки информации [2 – 4]. Однако данному формализму так же присущи некоторые недостатки, не позволяющие полно и глубоко исследовать механизмы распределения ресурсов и поведение системы в условиях конфликта доступа к ресурсам [5]. Следовательно, необходимо рассмотреть возможность расширения аппарата сетей Петри с целью адекватного описания процессов доступа к ресурсам различного типа. Так же необходимо разработать механизмы позволяющие обнаруживать конфликтные ситуации при доступе к ресурсам.

3. Основная часть

Для построения математической модели распределенной обработки информации рассмотрим конечное множество R, элементы которого соответствуют информационным ресурсам. При этом будем считать, что это множество является дизъюнктным объединением $R = S \coprod E$ двух подмножеств: S, элементы которого соответствуют информационным ресурсам с общим доступом, и E, элементы которого соответствуют информационным ресурсам с монопольным доступом. Рассмотрим также конечное множество процедур обработки информации, которое мы обозначим T.

Между информационными ресурсами и процедурами обработки данных существует два отношения:

- $\mathbf{IT} \subset R \times T$ – отношение, моделирующее набор утверждений «ресурс является входным для процедуры»;
- $\mathbf{TO} \subset T \times R$ – отношение, моделирующее набор утверждений «процедура модифицирует ресурс».

Таким образом, статическая структура предлагаемой модели задается пятеркой $\langle S, E, T, \mathbf{IT}, \mathbf{TO} \rangle$. На основе предложенной статической модели можно ввести следующие отображения:

- $t \mapsto t^* = \{r \in R \mid (t, r) \in \mathbf{TO}\} \subset R$ – множество выходных данных процедуры t;
- $t \mapsto t^+ = \{r \in R \mid (r, t) \in \mathbf{IT}\} \subset R$ – множество входных данных процедуры t;
- $r \mapsto r^* = \{t \in T \mid (r, t) \in \mathbf{IT}\} \subset T$ – множество процедур, использующих ресурс r как входные данные;
- $r \mapsto r^+ = \{t \in T \mid (t, r) \in \mathbf{TO}\} \subset T$ – множество процедур, модифицирующих ресурс r.

Для описания мгновенного состояния модели будем использовать отображение $s \in \{0,1\}^R$, которые в соответствии с терминологией сетей Петри назовем разметками. При этом равенство $s(r) = 1$ соответствует утверждению: «информационный ресурс был изменен, но это изменение еще не было обработано».

Динамика модели теперь может быть задана оператором на множестве разметок. Для определения оператора динамики нам потребуется ряд технических определений.

Процедуру $t \in T$ будем называть активной для разметки s, если $s \mid t^+ \equiv 1$. Множество активных процедур для разметки s будем обозначать $\mathrm{active}(s)$.

Содержательно, процедура является активной, если все входные данные для нее были изменены и еще не обработаны.

Процедуру $t \in T$ будем называть заблокированной для разметки s, если она активна для этой разметки и $\left(\exists r \in E \bigcap t^+\right)\left(\exists t_1 \in r^* \mid t_1 \neq t\right) s \mid t_1^+ \equiv 1$. Множество заблокированных процедур для разметки s будем обозначать $\mathrm{dead}(s)$. Содержательно, активная процедура является заблокированной, если для нее существует монопольный входной ресурс, который также является входным для другой активной процедуры.

В силу определения очевидно, что $\mathrm{dead}(s) \subset \mathrm{active}(s)$.

Это соотношение приводит к следующему определению.

Процедуру $t \in T$ будем называть эффективной для разметки s, если $t \in \mathrm{active}(s) - \mathrm{dead}(s)$. Множество эффективных процедур для разметки s будем обозначать $\mathrm{efficient}(s)$.

Таким образом, с каждой разметкой s связано подмножество активных процедур, разбитое на пару непересекающихся подмножеств: заблокированных и эффективных процедур.

Определим теперь множество информационных ресурсов, которые будут обрабатываться при заданной разметке s: $\mathrm{IN}(s) = \bigcup\limits_{t \in \mathrm{efficient}(s)} t^+$. Аналогично определим множество информационных ресурсов, которые будут изменены при заданной разметке s: $\mathrm{OUT}(s) = \bigcup\limits_{t \in \mathrm{efficient}(s)} t^*$.

Оператор динамики модели D определяется по формуле:

$$(Ds)(r) = \begin{cases} s(r), & r \notin \mathrm{IN}(s) \bigcup \mathrm{OUT}(s) \\ s(r), & r \in \mathrm{IN}(s) \bigcap \mathrm{OUT}(s) \\ 0, & r \in \mathrm{IN}(s) - \mathrm{OUT}(s) \\ 1, & r \in \mathrm{OUT}(s) - \mathrm{IN}(s) \end{cases}. \qquad (1)$$

Содержательно изменение данных информационных ресурсов моделируется следующим образом:

- если информационный ресурс не является входом или выходом ни одной эффективной процедуры, то его разметка не меняется;
- не меняется также разметка тех информационных ресурсов, которые являются одновременно входами каких-то эффективных процедур и выходами, возможно, других эффективных процедур (обработка изменения данных произведена, но возникли новые изменения в процессе этой обработки);
- разметка тех информационных ресурсов, которые являются входами каких-то эффективных процедур, но не являются выходами никаких эффективных процедур, устанавливается в нуль (обработка изменения

данных произведена и в ее ходе не возникло новых изменений рассматриваемых информационных ресурсов);

– разметка тех информационных ресурсов, которые являются выходами каких-то эффективных процедур, но не являются входами никаких эффективных процедур, устанавливается в единицу (в результате обработки изменений данных возникли новые изменения данных рассматриваемых информационных ресурсов).

Таким образом, предлагаемая модель является ординарной сетью Петри с модифицированной динамикой. С точки зрения авторов, определяемая оператором D динамика сети является адекватной логической моделью поведения вычислительной системы с распределенной обработкой данных. Эту модель мы будем называть далее сетью Петри с модифицированной динамикой или сокращенно – СПМД.

Естественной структурой на множестве разметок сети Петри с модифицированной динамикой является отношение частичного порядка:

$$s' \le s'' \Leftrightarrow (\forall r \in R)\left(s'(r) = 1 \Rightarrow s''(r) = 1\right). \qquad (2)$$

Утверждение 1. Если $s' \le s''$, тогда $\text{active}(s') \subset \text{active}(s'')$.

Действительно, если $t \in \text{active}(s')$, то $s' \mid t^+ \equiv 1$, но тогда и $s'' \mid t^+ \equiv 1$, а значит $t \in \text{active}(s'')$.

Утверждение 2. Если $s' \le s''$, тогда $\text{dead}(s') \subset \text{dead}(s'')$.

Действительно, если $t \in \text{dead}(s')$, то $s' \mid t^+ \equiv 1$. Следовательно, для некоторого $r_0 \in t^+ \bigcap E$ найдется $t_1 \in r_0^*$, для которого $s' \mid t_1^+ \equiv 1$. Тогда для этих же t, t_1, r_0 выполняется $s'' \mid t^+ \equiv 1$ и $s'' \mid t_1^+ \equiv 1$, а это и означает, что $t \in \text{dead}(s'')$.

Модификация динамики сети Петри ориентирована на выявление конфигураций событий обработки данных, которые приводят к блокировке анализируемой системы.

Систему будем считать заблокированной, если $\text{active}(s) \neq \varnothing$ и $\text{dead}(s)=\text{active}(s)$. Следовательно, если в результате работы модели на каком-то из шагов окажется, что система в заблокированном состоянии, то она не сможет дальше продолжать свою работу: $\text{efficient}(s) = \varnothing$, т.е ни одна из процедур $t \in T$ не сможет выполниться.

На каждом шаге работы модели формируются множества $\text{dead}(s)$ и $\text{efficient}(s)$. Процедуры, принадлежащие множеству $\text{efficient}(s)$, могут быть выполнены на дальнейших шагах работы системы. Процедуры, принадлежащие множеству $\text{dead}(s)$, являются заблокированными и не могут

быть выполнены при дальнейшей работе модели. Следовательно, множество процедур $t \in \mathrm{dead}(s)$ является потерей функциональности системы.

В этой связи интересным является вопрос о возможности выявить данные ситуации без проведения имитационного моделирования, что позволит с одной стороны сэкономить время анализа, а с другой – однозначно выявить последовательности выполнения процедур t, приводящие к блокировке.

Естественным методом решения данной задачи является анализ дерева достижимости.

Изменение динамики функционирования сети Петри, а так же требование к ординарности приведет к модификации алгоритма построения дерева достижимости построенного в [2]. Напомним, что вершины дерева достижимости помечены маркировками. Для формализации алгоритма построения покрывающего дерева введем ряд уточняющих понятий:

1. Терминальной вершиной покрывающего дерева называется вершина, для которой $\mathrm{efficient}(s) = \varnothing$. Множество терминальных вершин V_T является дизъюнктным объединение двух подмножеств: $V_T = V_E \coprod V_D$. Множество V_E содержит вершины, для которых $\mathrm{active(s)} = \varnothing$. Множество V_D содержит вершины, для которых $\mathrm{dead(s)} = \mathrm{active(s)}$, следовательно, система не может продолжать работу по причине блокировки информационных ресурсов.

2. Внутренняя вершина – вершина, обработанная алгоритмом. Множество внутренних вершин V_V также является дизъюнктным объединение двух подмножеств: $V_V = V_A \coprod V_B$. Множество V_A содержит вершины, для которых $\mathrm{efficient}(s) = \mathrm{active}(s)$. Множество V_B содержит вершины, для которых: $\mathrm{dead(s)} = \mathrm{active(s)} - \mathrm{efficient}(s) \neq \varnothing$.

Модифицированный алгоритм построения дерева достижимости заключается в следующем.

Построение дерева достижимости начинается с начальной разметки. Начальная разметка определяет граничную вершину, алгоритм обрабатывает только граничные вершины.

Для каждой граничной вершины v выполняется следующий набор действий.

1. Определяется множество $\mathrm{active(s)}$, а так же его составляющие $\mathrm{efficient}(s)$ и $\mathrm{dead(s)}$.

2. Если $\mathrm{efficient}(s) = \varnothing$, то вершина является терминальной. При этом, если $\mathrm{active(s)} = \varnothing$, то вершина включается в множество V_E, в противном случае в множество V_D.

3. Если $\text{efficient}(s) \neq \varnothing$, для данной вершины анализируется маркировка s: в случае если в дереве существует вершина V_V для которой: $vv_i(s)=v(s)$, то данная вершина v объявляется дублирующей. В противном случае для каждой процедуры $t \in \text{efficient}(s)$ создается новая вершина v' дерева достижимости. Разметка новой вершины определяется в соответствии с (1).

4. Строится дуга с пометкой выполненной процедуры t_i, направленная от вершины v к вершине v'. Вершина v становится внутренней и включается в множество V_A, если $\text{efficient}(s)=\text{active}(s)$ и в множество V_B в противном случае. Вершина v объявляется граничной.

Алгоритм останавливает работу, когда в дереве не остается граничных вершин.

В результате построения покрывающего дерева сети Петри определяется набор вершин, принадлежащих множествам V_D и V_B. Вершины $v_i \in V_D$, соответствуют состояниям системы, при которых она не может продолжать работу по причине блокировки всех процедур. Вершины $v_i = V_B$, соответствуют состояниям системы, при которых она утрачивает часть функциональности (некоторые процедуры уже не смогут выполниться).

5. Выводы

В работе показана возможность модификации аппарата сетей Петри. Сеть Петри с модифицированной динамикой позволяет адекватно описать функционирование распределенной системы обработки информации использующей два типа ресурсов (ресурсы с монопольным доступом и ресурсы с общим доступом). Алгоритм построения дерева достижимости для СПДМ позволяет выявить конфигурации событий системы, приводящие к потере функциональности. Таким образом, использование предложенных в работе модификаций сетей Петри, которые включают алгоритм построения дерева достижимости, открывает путь к повышению эффективности анализа распределенных систем обработки информации.

Список литературы: ***1.*** Высокопроизводительные параллельные вычисления на кластерных системах. Материалы международного научно-практического семинара. // *Под ред. проф. Стронгина Р. Г.* Н. Новгород: Изд-во Нижегородского ун-та, 2002, 217 с. ***2.*** *Питерсон Дж.* Теория сетей Петри и моделирование систем. Пер. с англ. – М.: Мир, 1984. –264 с. ***3.*** *Таненбаум Э., ван Стен М.* Распределенные системы. Принципы и парадигмы. – СПб.: Питер, 2003. – 877 с. ***4.*** *Башкин В. А., Ломазова И. А.* Подобие обобщенных ресурсов в сетях Петри // Труды второй Всероссийской научной конференции – М.: Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, 2005. – С. 330-337. ***5.*** *Востокин С. В.* Метод описания пространственно распределенных параллельных процессов. – СНЦ РАН, 2004. – 84 с.

Поступила в редколлегию 03.04.06